

Express Mail Label No. EL 823 500 355 US

**APPLICATION FOR LETTERS PATENT
OF THE UNITED STATES**

NAME OF INVENTOR(S):

Marc A. Konas
3433 Plow Road
Mohnton, PA 19540
UNITED STATES OF AMERICA

TITLE OF INVENTION:

A System, Method, and Article of Manufacture for Creating and Updating an Application Using Software Application Elements

TO WHOM IT MAY CONCERN, THE FOLLOWING IS
A SPECIFICATION OF THE AFORESAID INVENTION

105201 4405201

U.S. Express Mail Label No. EL 823 500 355 US

**A SYSTEM, METHOD, AND ARTICLE OF
MANUFACTURE FOR CREATING AND UPDATING
AN APPLICATION USING SOFTWARE APPLICATION ELEMENTS**

BACKGROUND OF THE INVENTIONS

Software is often corrected or augmented after and where it is installed. Many times, these updates are applied as patches which modify only predetermined portions of the software.

However, as the augmentation and/or number of patches grows, the ability to manually apply these changes in a timely and accurate manner degrades. At times, installations may use scripts to automate or otherwise assure a correct application of the change.

Over the last several years, software applications have increased in complexity such as those that have been implemented and developed using a web deployable client/server software paradigm, including such functions as accounting, human resources, distribution and materials management application software. Lawson Associates, Inc. of St. Paul, Minnesota creates and markets illustrative enterprise software solutions under the LAWSON SOFTWARE trademark that enhance financial, human resources and supply chain management. See <http://www.lawson.com/>. Oracle Corporation of San Francisco, California, SAP AG of Walldorf, Germany, and PeopleSoft Inc. of Pleasanton, California also manufacturer and market similar applications.

SUMMARY

The present inventions comprise a system, method, and article of manufacture for use in updating a target application with a plurality of software application data elements. In an exemplary embodiment, the system comprises a source of a plurality

of application data elements suitable for updating a plurality of different target applications; a first processor for identifying and extracting selected application data elements from the source in response to user selection of a target application; and an output processor for mapping the extracted selected application data elements to
5 corresponding file locations for replacement of corresponding data elements using mapping information associating the selected application data elements with file location data of the corresponding data elements.

The present inventions may work in cooperation with numerous software applications, by way of example and not limitation including enterprise planning
10 software such as that provided by Lawson Associates, Inc., Oracle Corporation of San Francisco, California, SAP AG of Walldorf, Germany, and PeopleSoft Inc. of Pleasanton, California.

However, the scope of protection is not limited by the summary of an exemplary embodiment set out above, but is only limited by the claims.

15 **BRIEF DESCRIPTION OF THE DRAWINGS**

These and other features, aspects, and advantages of the present inventions will become more fully apparent from the following description, appended claims, and accompanying drawings in which:

Fig. 1 is a schematic of an exemplary system of the present inventions;

20 **Fig. 2** is an exemplary selection menu; and

Fig. 3 is a block diagram of an exemplary embodiment of the present inventions.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In general, throughout this description, if an item is described as implemented
25 in software, it can equally well be implemented as hardware. Further, although an

enterprise wide software environment is described herein as an exemplary embodiment, the present inventions are not limited to such a software environment and are limited, instead, by the claims herein below.

Referring now to **Fig. 1**, a schematic representation of an exemplary embodiment, in this description target software applications will be identified by the application processors 32, 34, 36 in which those target software applications execute, i.e. by "32a, 34a, 36a." As further used herein, each target software application 32a, 34a, 36a may be different from others of the software applications 32a, 34a, 36a, have at least some software elements in common, or be the same application.

Source 5 comprises data resident or otherwise accessible to first processor 10, e.g. files on one or more persistent data stores 12. Data at source 5 further comprise a plurality of application data elements 6. Application data elements 6 may comprise individual files and/or file libraries that describe or contain totally new software, new functionality for existing software, patches to existing software, or a combination thereof. Additionally, application data elements 6 may be suitable for updating one or more target software applications 32a, 34a, 36a executing in application processors 32, 34, 36. Application processors 32, 34, 36 may be located remotely from first processor 10. Additionally, source 5 may be a single source file or may comprise a plurality of source files.

Identification software executing in first processor 10 identifies and extracts desired application data elements 6 from source 5 such as in response to the user's selection of target software applications 32a, 34a, 36a via a menu such as menu form 200 (**Fig. 2**). In a currently envisioned alternative embodiment, identification software may additionally be triggered by external events, by way of example and not

limitation comprising timers, flags, semaphores, scripts, triggers, or the like, or combinations thereof; triggered manually; or triggered by a combination thereof.

The extracted application data elements 6 may comprise a plurality of sets of selected application data elements 6 selected in response to user selection of a
5 corresponding plurality of target software applications 32a, 34a, 36a. Selection criteria may be based on application processor 32, 24, 26 and/or customer parameters. Additionally, the extracted application data elements 6 may comprise a plurality of sets of programmatically selected application data elements 6 that are selected because they are associated with the corresponding plurality of target software
10 applications 32a, 34a, 36a.

The extracted selected application data elements 6 will be used to replace corresponding data elements 6 existing in current software application 32a, 34a, 36a. Output processor 30 may be used to map the extracted selected application data elements 6 to corresponding file locations such as at data store 40. Output processor
15 30 may use mapping information associating the selected application data elements with the file locations of corresponding data elements 6, by way of example and not limitation including customer configuration data 14 and software picklists 16. Additionally, output processor 30 may copy the extracted selected application data elements 6 to their corresponding file locations at application processors 32, 34, 36.
20 In exemplary embodiments, first processor 10 and output processor 30 may be different computers or may be different software processes executing in the same computer, e.g. first processor 10.

Authorization software, which may be software executing at output processor 30 or an additional processor, by way of example and not limitation such as computer
25 20, may be present to inhibit replacement of corresponding data elements 6 in

response to predetermined criteria. In an exemplary embodiment, the criteria comprise at least one of billing information, user identification codes, user passwords, detection of a data element error, capabilities of processor 32, 34, 36, software environments such as at processor 32, 34, 36, and the like, or combinations thereof. In such an environment, the authorization software will act as a gatekeeper, allowing selected target computers, e.g. 32, that meet predetermined authorization criteria to receive and install data elements 6 while keeping other potential target computers, e.g. 34, that do not meet the predetermined authorization criteria to be excluded from the update.

Referring now to **Fig. 2**, an exemplary selection menu, a menu generator (not shown in **Fig. 1**) may be used to initiate a display of a menu 200 such as at first processor 10 (**Fig. 1**). The menu generator may query or otherwise determine available data in source 5 as well as available customers and application processors 32,34,36 and data regarding the customers and application processors 32,34,36 to determine which updates are available as well as which target applications 32a,34a,36a are available to receive the updates. The menu generator may then configure presentation menus accordingly, such as menu 200.

Menu 200 may be used to allow a user with appropriate authority to select target applications 32a, 34a, 36a. These selections may be based on predetermined selection criteria which may include at least one of target application identification codes, target application version codes, and target application geographical associations. In an exemplary manual mode of selection using a menu shown as an electronic form 200, a user may indicate a desired set of selections such as by using menu bars 210 and/or dialogs 220. Electronic form 200 may further comprise options to allow additional selection criteria data. By way of example and not limitation,

these criteria may include application or environment criteria such as billing information or support level information. At least a portion of these criteria may be received from remote source 20 such as through data communications links such as through the Internet 100 (shown in **Fig. 1**).

5 Electronic menu form 200 may take many forms, as will be familiar to those of ordinary skill in the software arts. In **Fig. 2**, electronic form 200 comprises one or more drop-down selections from menu bar 210 and dialog boxes 220 that allow a user to specify selection criteria data. By way of example and not limitation, dialog 220a may be present to allow a user with appropriate authority to designate one or more
10 application data elements 6 to apply. As will be understood by those of ordinary skill in the computer arts, locations of source 5 (**Fig. 1**) may be identified such as with menu 210, dialog 220a, or a combination thereof. In like fashion, dialog 220b, or any dialog 220, may be present or invoked such as from menu bar 210, function keys (not shown in the figures), hot regions, buttons, or the like, to provide for selection of one
15 or more customers. Dialog 220c may also be present to select one or more targets 32, 34, 36 (**Fig. 1**) for application data elements 6 (**Fig. 1**) selected. As will be familiar to those of ordinary skill in the art, other application or environment criteria such as billing information or support level information may be presented, such as in response to a dialog 220 or menu 210 selection.

20 **In the operation of an exemplary embodiment**, referring now to **Fig. 1** and **Fig. 3**, a block diagram of an exemplary embodiment of the present inventions, one or more application data elements 6 (**Fig. 1**) to be applied to a targeted software application 32a, 34a, 36a (**Fig. 1**) resident in a application processor such as 32, 34, 26 (**Fig. 1**) are identified 300 and then extracted, collected, and gathered 310. A
25 script, or other utility, function, or module as will be familiar to those of ordinary skill

in the software arts, may be used to provide an ability to identify or otherwise extract selected application data elements 6 from source 5.

The gathered application data 6 (**Fig. 1**) may be further placed into software packets 320 (**Fig. 3**) such as at first computer 10 (**Fig. 1**) or output processor 30 (**Fig. 1**), e.g. at step 330 (**Fig. 3**). The targeted software applications 32a, 34a, 36a (**Fig. 1**) may further comprise an integrated utility for applying such packets or may have an associated utility to apply such packets locally to the targeted software applications 32a, 34a, 36a (**Fig. 1**). A user may indicate a desired set of selections such as by using electronic menu form 200 (**Fig. 2**). .

Once selected and collected, selected application data elements 6 may be copied and stored 330 (**Fig. 3**) to a separate, predetermined storage location such as persistent data store 40 (**Fig. 1**). By way of example and not limitation, in some software environments scripting languages may be used to create software picklists 16 (**Fig. 1**) to extract application data elements 6 (**Fig. 1**) and copy them to appropriate locations on a server, e.g. processor 30 (**Fig. 1**). In currently envisioned alternative embodiments, the predetermined storage location may be an application service provider (not shown in the figures) as that term will be familiar to those of ordinary skill in the Internet arts.

Once the software application data elements 6 (**Fig. 1**) are placed into an appropriate storage location, application software update functions may be triggered 344 (**Fig. 3**) such as at application processors 32,34,36 (**Fig. 1**) for local installation 350 (**Fig. 3**). Additionally, authorization software, e.g. software executing at output processor 30 (**Fig. 1**), may selectively permit triggering of application software update functions only at those computers that meet the update criteria, e.g. step 342 (**Fig. 3**).

In an exemplary embodiment, the update functions may further be capable of invoking various external processes that may be required during the software update that are not part of the update functions, by way of example and not limitation in an exemplary embodiment start and stop commands for the one or more software applications 32a, 34a, 36a (**Fig. 1**) to be updated. In the preferred embodiment, application of the selected application data elements 6 (**Fig. 1**) is automated as automation lends itself to a more accurate and more complete installation of the application data elements 6 (**Fig. 1**) in their correct order.

In a currently envisioned alternative embodiment, the present inventions may be used to aid in creating application software with adaptively selectable functions. In such an embodiment, source 5 (**Fig. 1**) comprises a plurality of application data elements 6 (**Fig. 1**) corresponding to a plurality of functional elements. In this exemplary embodiment, in response to user selection of target application functions, after first processor 10 (**Fig. 1**) identifies and extracts selected application data elements 6 (**Fig. 1**) from source 5 (**Fig. 1**) such as a steps 300 and 310 (**Fig. 3**), output processor 30 (**Fig. 1**) maps 340 (**Fig. 3**) the extracted selected application data elements 6 (**Fig. 1**) to corresponding file locations at application processors 32, 34, 36 (**Fig. 1**). Using mapping information that associates the selected application data elements 6 (**Fig. 1**) with file location data of the corresponding data elements, when the targeted applications 32a, 34a, 36a (**Fig. 1**) are updated, new or modified target software applications 32a, 34a, 36a (**Fig. 1**) are thus created that comprise the user selected functions.

It will be understood that various changes in the details, materials, and arrangements of the parts which have been described and illustrated above in order to explain the nature of these inventions may be made by those skilled in the art without

departing from the principle and scope of the inventions as recited in the following claims.

100230518-102501